MathMatrix

• W.r.t. Style:
  • Make instance variables private
  • Use an auto-formatter!
  • Watch out for unnecessary repeated computations inside of loops

• W.r.t Experiments:
  • Answer all the questions!
  • It’s ok if your timing data doesn’t support your Big O analysis of a method! Remember, Big O is a big simplification!
Formatting

- Again, many style issues can be avoided by using your IDE’s auto-formatter:

<table>
<thead>
<tr>
<th>IDE</th>
<th>Windows/Linux</th>
<th>macOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IntelliJ</td>
<td>Ctrl+Alt+L</td>
<td>Command+Option+L</td>
</tr>
<tr>
<td>Eclipse</td>
<td>Ctrl+Shift+F</td>
<td>Command+Shift+F</td>
</tr>
<tr>
<td>VSCode</td>
<td>Shift+Alt+L</td>
<td>Shift+Option+F</td>
</tr>
</tbody>
</table>
Section Problem

Polymorphism & Inheritance
Name Surfer

Due Thursday, February 10
Advice for NameSurfer

• On the assignment page, you’ll see there are certain conditions where you’ll want to skip a line of input

• If a line of input is to be skipped, DON’T create a NameRecord for it! That wastes space!

• See Piazza Post @247 (it’s pinned)
Advice for **NameSurfer**

```java
public static NameRecord buildRecord(String input, /* ... */) {
    // Check conditions for input
    if (inputIsValid) {
        // You can make constructor private so that this
        // is the only way of making NameRecord objects
        return new NameRecord(/* ... */);
    } else {
        return null;
    }
}
```
Advice for **NameSurfer**

- Be VERY careful of magic numbers
  - Especially 10, 1000, 1001
  - All numbers other than -1, 0, 1, 2 OR which have a special meaning should be constants

- Sort your data as few times as possible
  - Sorting is (relatively) slow, don’t do it more than you have to!

- And, of course, use your auto-formatter!
Extra Problem

GenericList & NameRecords